Computer Fundamentals

Lecture 5

Memory System Design (part I)

Asst. Lect. Ali Al-khawaja

Class Room

Learning Outcomes

By the end of this lecture, students should be able to:

01

Identify the three main components of a computer system: CPU, Memory, and I/O.

02

Explain how these components communicate using an interconnection structure (bus/interconnect).

03

Correctly interpret common measurement units (data size, frequency, time, throughput).

04

Define embedded systems, provide examples, and explain why some systems require real-time constraints.

05

Explain the relationship between IoT and cloud computing and why cloud platforms are widely used.

Lecture Contents

1

Introduction

2

Computer Components

3

Measurement Units in Computing

4

Embedded Systems

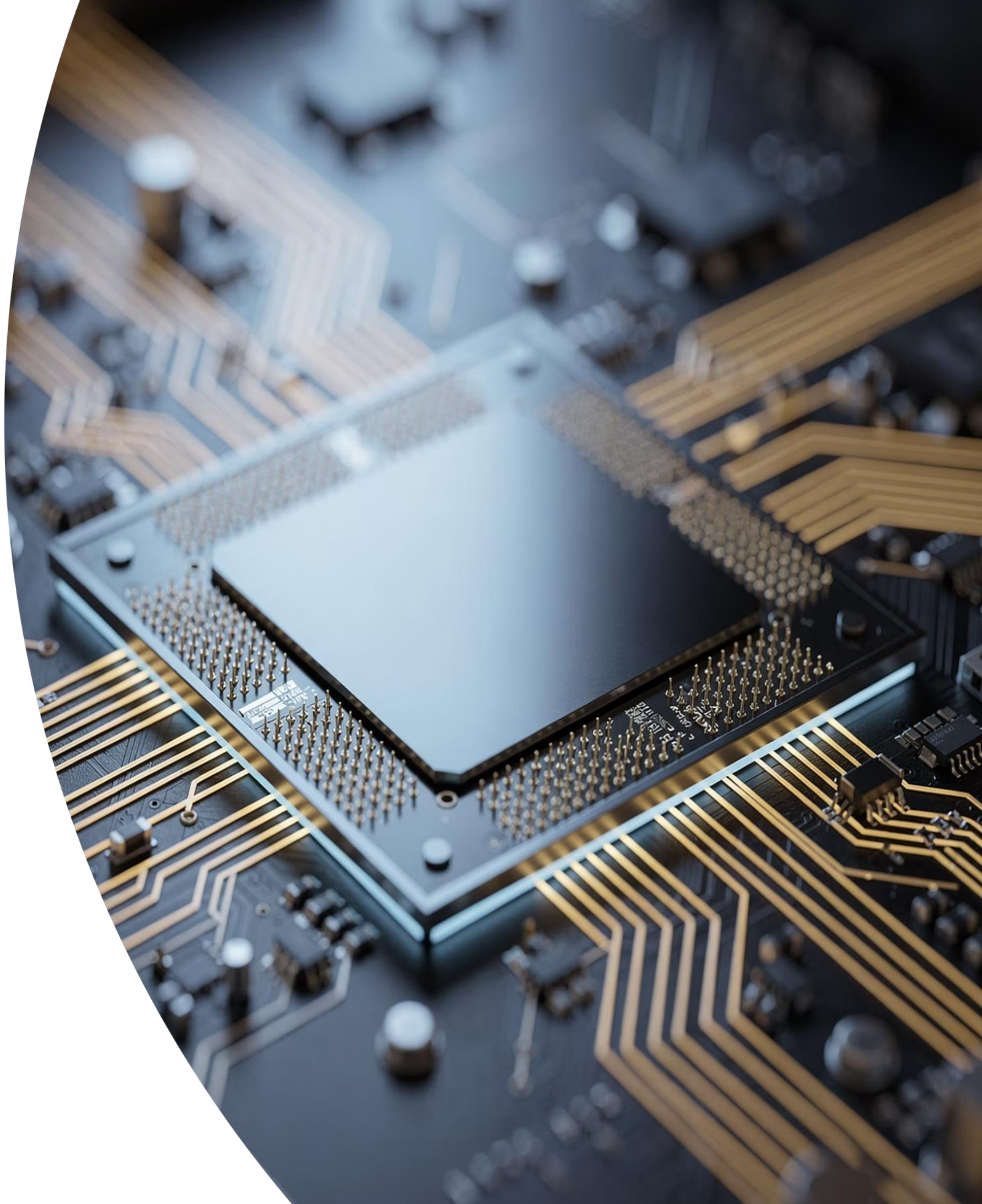
5

IoT and Cloud Computing

Why "Performance" is more than just CPU speed?

When people hear "computer performance," they often think only about the CPU and its clock speed. In reality, performance depends on how well the entire system works together. A computer is built from multiple components that continuously exchange data. If one component is slow—or if the connection between components is limited—overall performance can drop significantly. A system can have a fast CPU but still feel slow because of:

- slow memory access,
- slow input/output devices (storage, network, peripherals),
- limited data transfer capability between components.



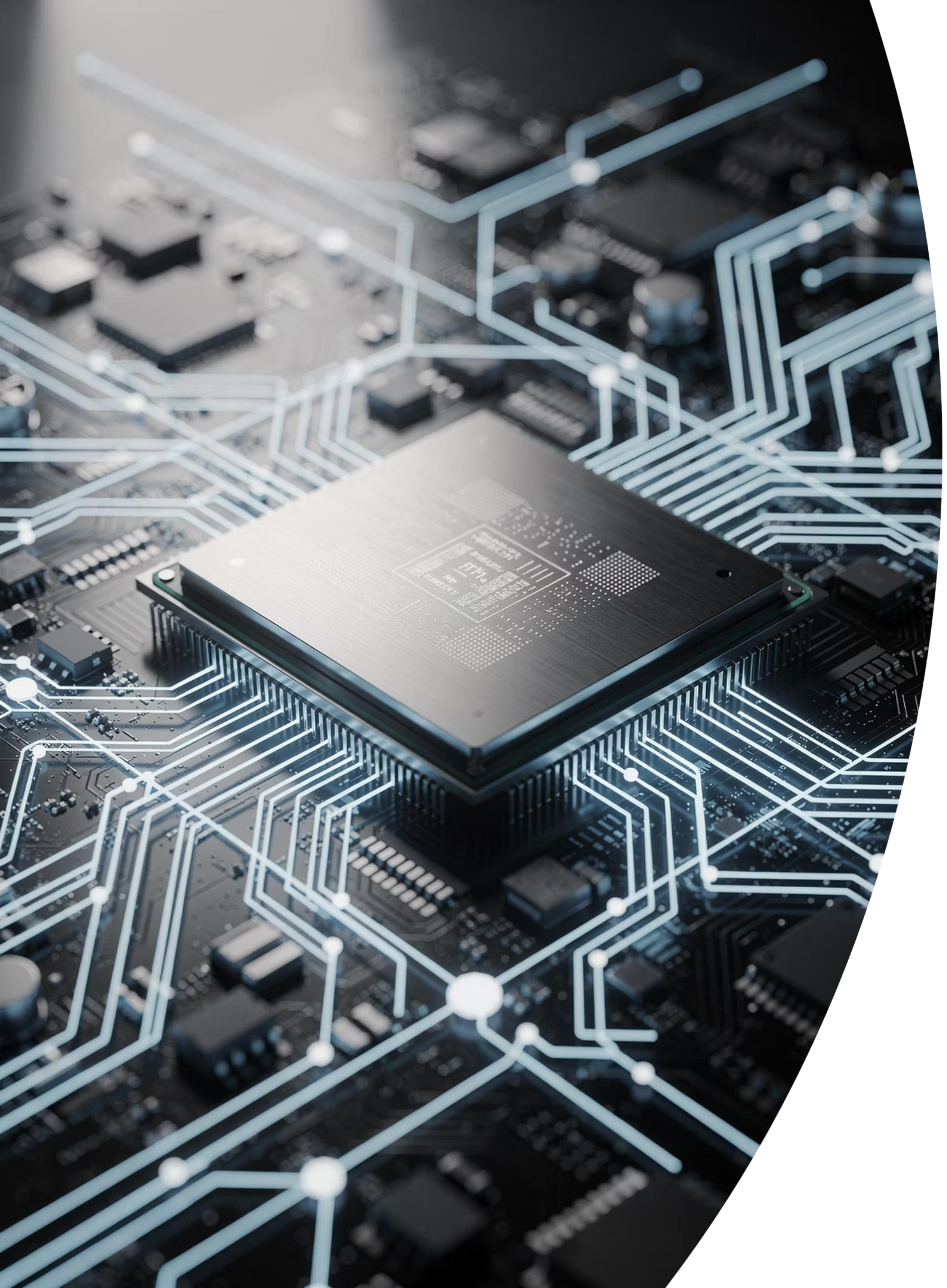
Computer Components

A computer system is not a single device that "works alone." It is a coordinated set of components that cooperate to execute programs and deliver results to the user. Each component has a specialized role: the processor performs computations and controls the system, memory stores instructions and data needed during execution, and input/output modules connect the system to external devices such as storage, keyboard, screen, sensors, and networks. These components continuously exchange data, and their coordination determines the overall behavior and performance of the computer.

A typical computer system is built from three fundamental components:

1. **CPU (Processor)**
2. **Memory**
3. **Input/Output (I/O)**

In addition, these components communicate through an Interconnection Structure (bus/interconnect), which enables data transfer between them.



1) CPU (Processor)

The CPU is responsible for:

- reading instructions and data,
- executing operations,
- writing results,
- coordinating system operation through control signals.

> The CPU is the "***manager***" that decides what to do next, but it needs memory and I/O to actually get work done.

2) Memory

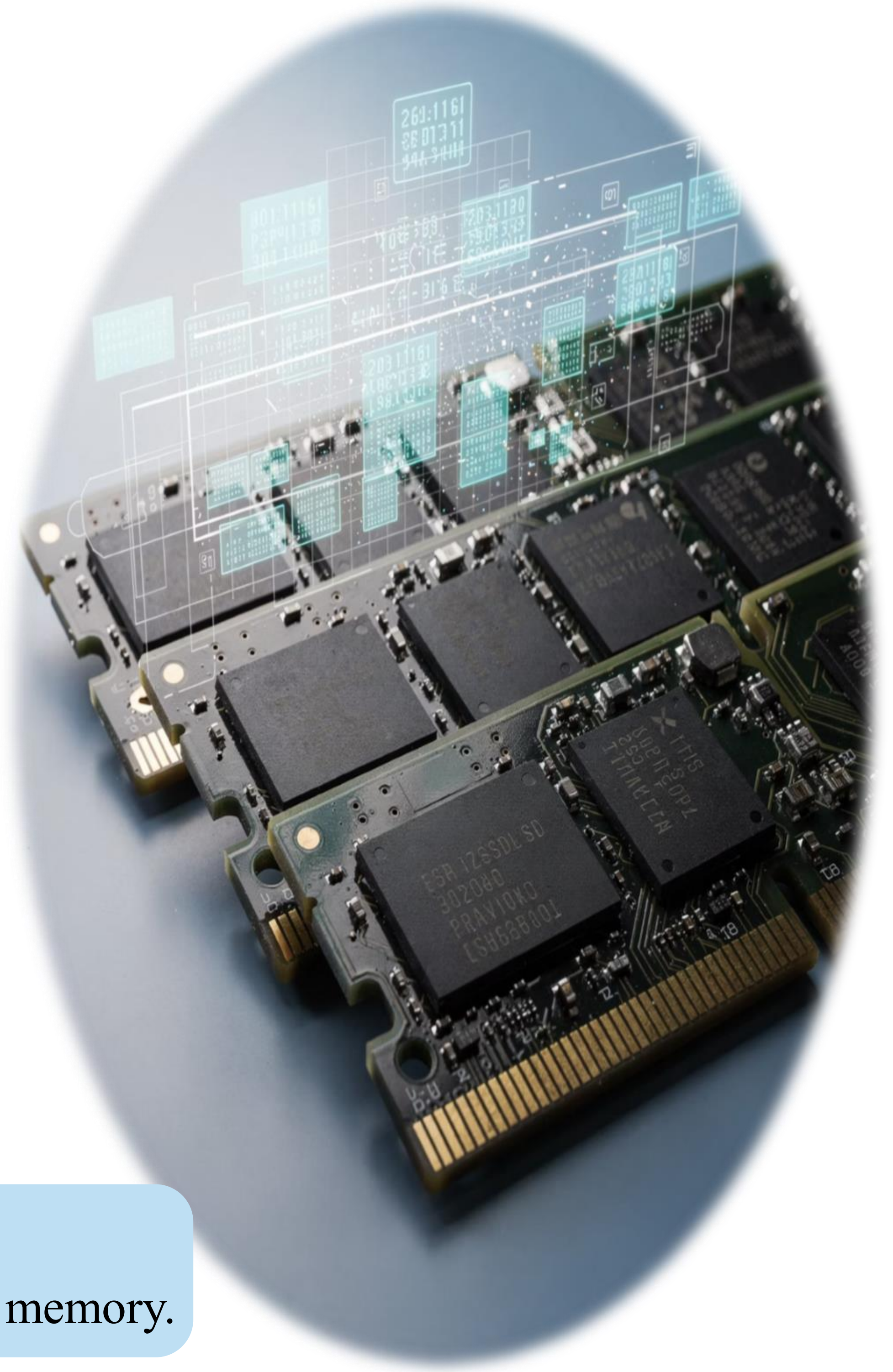
Memory stores:

- program instructions currently being used,
- data being processed,
- intermediate results while tasks are running.

Addressing

Memory can be imagined as many storage locations (like numbered boxes).

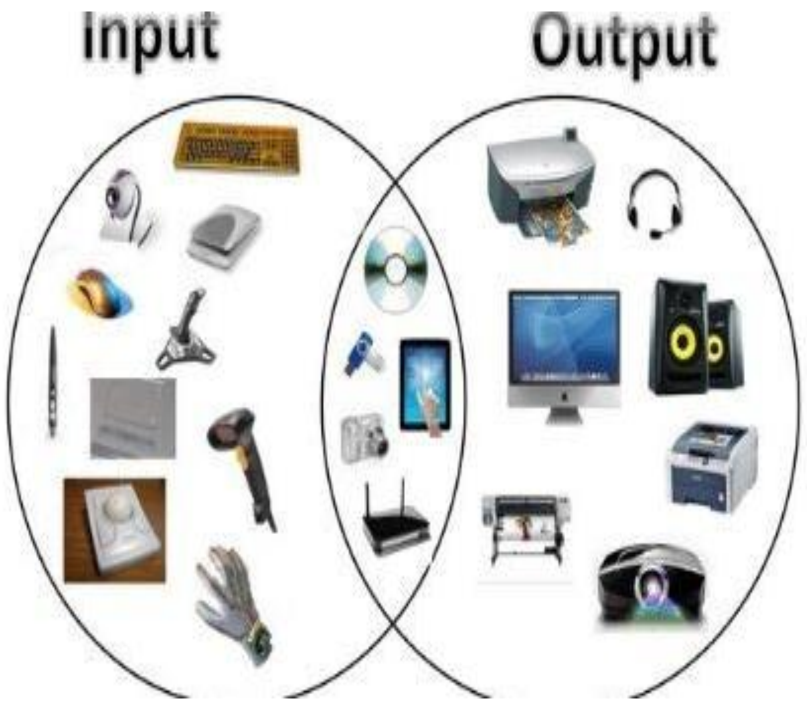
- The number is the **address**
- The content is the **data**



Why memory affects performance?

Even if the CPU is fast, it cannot work efficiently if it must wait frequently for data from memory.

3) Input/Output (I/O)



I/O is how the system interacts with the outside world:

User Interface

keyboard, mouse, screen

Storage Devices

HDD/SSD/USB

Network Devices

Wi-Fi/Ethernet

Peripherals

sensors, cameras, printers, etc.

Interrupt idea

Sometimes an I/O device "signals" the CPU that something needs attention (for example: new data arrives from the network).

Why I/O affects performance?

I/O devices are usually slower than CPU and memory. Many systems therefore use buffering and specialized mechanisms to reduce CPU waiting time.

Interconnection Structure: How components communicate?

CPU, memory, and I/O must exchange data through an interconnection structure (commonly a bus or a more modern interconnect). A computer must support basic data transfers such as:

1. **Memory → CPU**

   CPU reads instructions/data

2. **CPU → Memory**

   CPU writes results/data

3. **I/O → CPU**

   CPU reads data from a device

4. **CPU → I/O**

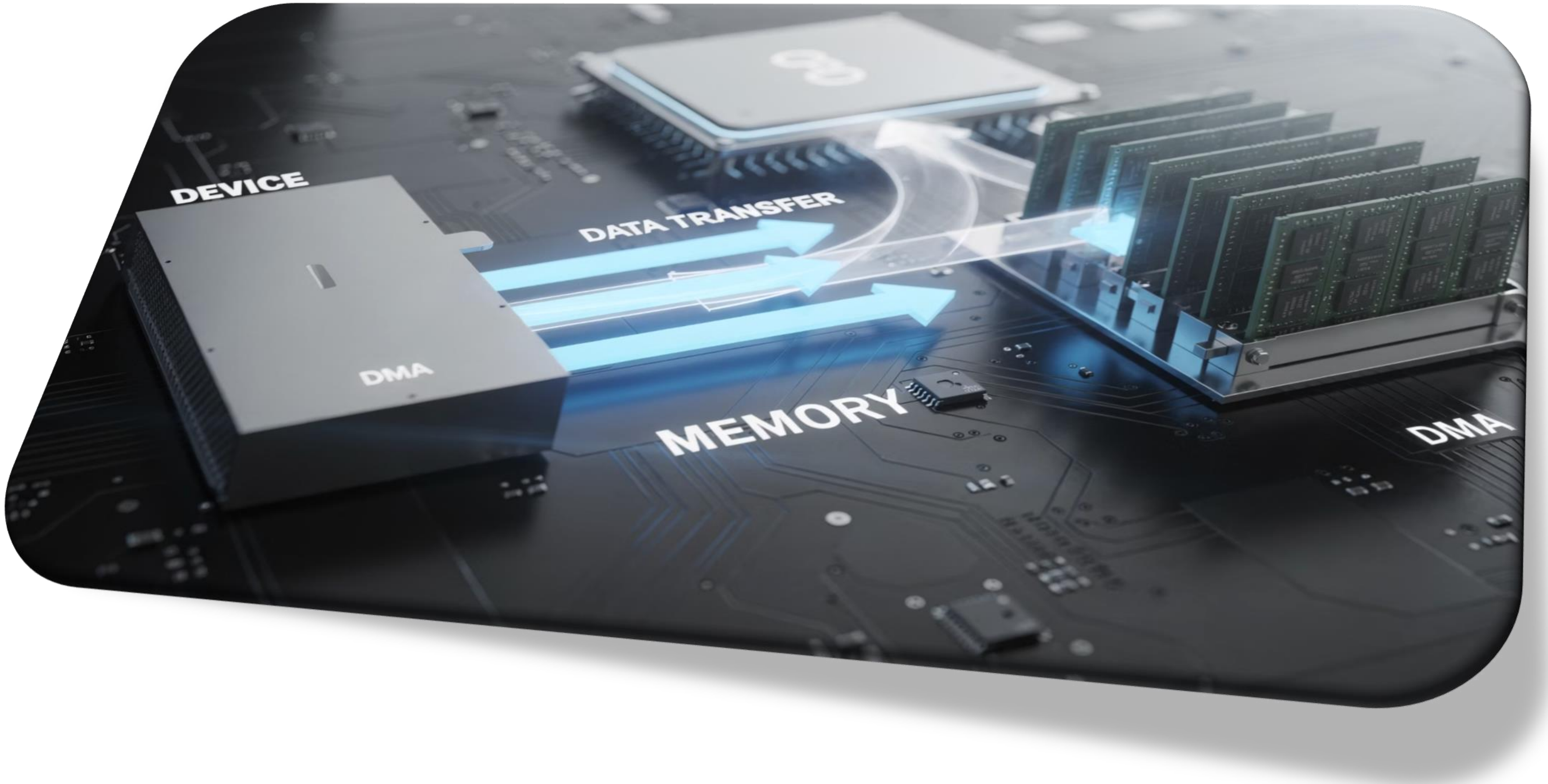
   CPU sends data to a device

5. **I/O ↔ Memory (DMA)**

   a device transfers data directly to/from memory without forcing the CPU to handle every byte

Why DMA (Direct Memory Access) important?

Without DMA, the CPU would spend a lot of time moving large blocks of data (for example: copying a file). DMA reduces CPU workload and can improve overall system efficiency.



Measurement Units

- In computer science, we describe hardware and system performance using measurement units. These units help us understand how much data a device can store, how fast it can process instructions, how long an operation takes, and how quickly data can be transferred.

- learning measurement units is essential because they appear in everyday technology specifications such as RAM size (GB), CPU speed (GHz), internet speed (Mbps), and response time (ms). Correct interpretation of these units is the foundation for comparing devices and estimating performance.

Main Measurement Unit Categories

The most important measurement units in computing can be grouped into four categories:

Data Size Units

(bit, byte, kB, MB, GB, TB)

Frequency Units (Clock Rate)

(Hz, kHz, MHz, GHz)

Time Units (Latency/Delay)

(ms, µs, ns)

Data Transfer Rate Units

(bps, Kbps, Mbps, Gbps)

Data Size Units (How much data?)

Data size units describe storage capacity and file size. They tell us how much information can be stored in memory or on a disk.

bit (b)

the smallest unit of information (0 or 1).

byte (B)

typically equal to 8 bits; it is the most common unit for file sizes.

kB, MB, GB, TB

larger units used for files, memory, and storage.

practical understanding:

- A simple text file may be in kB.
- A photo may be in MB.
- A phone storage or laptop drive may be in GB or TB.

Important note (common confusion):

Students may see size written as MB and speed written as Mb—these are not the same.

- **B = Byte**
- **b = bit**

And **1 Byte = 8 bits**.

Frequency Units (Clock Rate: How fast the CPU cycles?)

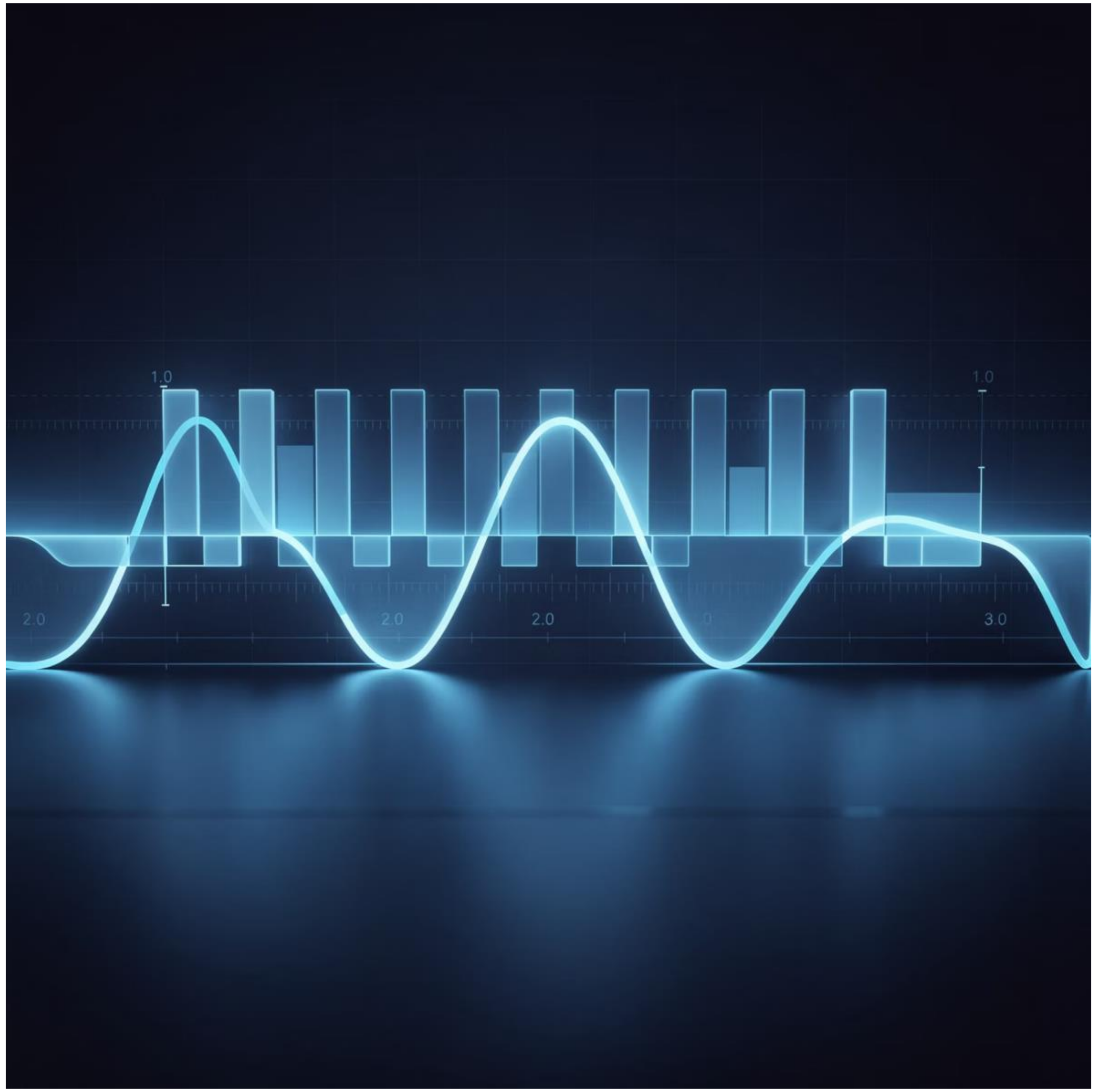
Frequency units describe how many cycles happen per second in a clocked system like a CPU.

Hz
cycles per second

kHz
thousand cycles per second

MHz
million cycles per second

GHz
billion cycles per second



practical understanding:

If a CPU runs at a higher GHz, it means its clock produces more cycles per second. However, overall performance does not depend on clock speed alone, because the CPU may still need to wait for memory or I/O.

Time Units (Latency/Delay: How long does it take?)

Time units describe how long an operation or response takes. These are important for understanding system responsiveness.

- **ms (millisecond)**
  10^{-3} seconds
- **µs (microsecond)**
  10^{-6} seconds
- **ns (nanosecond)**
  10^{-9} seconds

practical understanding:

- If something takes many ms, humans can often notice the delay.
- Very small delays in hardware operations may be described in µs or ns.

Data Transfer Rate Units (Throughput/Bandwidth: How fast data moves?)

These units describe how quickly data can be transferred between devices or across networks.

bps
bits per second

Kbps
thousand bits per second

Mbps
million bits per second

Gbps
billion bits per second

Critical point: Mbps vs MB/s

- **Mbps** = megabits per second (common in internet and network speeds)
- **MB/s** = megabytes per second (common in file transfers/storage speeds)

Conversion rule (very important):

Since 1 Byte = 8 bits, then:

MB/s ≈ Mbps ÷ 8

Example:

If your internet speed is 80 Mbps, then the maximum theoretical download speed is: 80 ÷ 8 = 10 MB/s
(Real speed is usually lower due to overhead and network conditions.)

Embedded Systems

What is an Embedded System?

An embedded system is a computer system built into a larger device to perform specific tasks. Unlike a laptop (general purpose), an embedded system is designed for a focused function, such as controlling a sensor, monitoring a process, or operating a device.

Real-life examples

- washing machines and microwaves
- printers and cameras
- routers and smart TVs
- cars (braking systems, engine control)
- medical devices (patient monitors, infusion systems)

Key characteristics

Embedded systems commonly prioritize:

low cost

low power consumption

reliability

small size

performing a dedicated task efficiently

Why timing important?

In many embedded systems, correctness includes timing:

- The system must produce the correct response within a required time.
- A late response can be considered a failure even if the logic is correct.

examples:

- A safety system must trigger fast enough to prevent harm.
- A medical monitoring system must detect dangerous conditions and alert quickly.

Microprocessor vs Microcontroller

Microprocessor

- Often associated with general-purpose computing platforms.
- Usually relies on external components for memory and I/O in many designs.

Microcontroller

- Often integrates CPU + memory + I/O interfaces into one chip.
- Commonly used in embedded applications.
- Typically operates at lower speeds than modern laptop CPUs, but is optimized for efficiency and control tasks.

Key point:

Embedded performance is often about "doing the right job efficiently," not about maximum GHz.

IoT and Cloud Computing

IoT (Internet of Things) refers to connecting large numbers of smart devices that can:

- sense the environment (sensors),
- communicate data,
- and sometimes act on the environment (actuators).

These devices are often embedded systems (small, specialized, and efficient).

Cloud Computing

Cloud computing provides services over networks such as:

data storage

computing power

applications and platforms

Instead of each device doing heavy processing locally, devices can send data to cloud services for:

storage

analysis

dashboards and monitoring

system updates and management

Why cloud is used with IoT?

Many IoT devices are limited in processing power and storage. The cloud is helpful because it can

- handle huge volumes of data
- scale to support many devices
- provide centralized monitoring and control

Quick Quiz

1. Name the three core components of a computer system.
2. What is an interconnection structure and why is it necessary?
3. Convert 64 Mbps to approximate MB/s.
4. Give two examples of embedded systems.
5. What does "real-time constraint" mean?
6. Why are cloud systems commonly used with IoT?

Homework

01

Write 8–10 lines explaining why performance depends on CPU, memory, I/O, and interconnect together.

02

Choose one embedded device and describe one timing requirement it might have.

03

Explain the difference between data size (GB) and throughput (Gbps), with one example.

04

Describe one benefit and one risk of relying on cloud computing.

Thank you…

Any questions??

My google site

يرجى مسح رمز الاستجابة السريعة QR Code لتعبئة نموذج التغذية الراجعة حول المحاضرة. ملاحظاتكم مهمة لتحسين المحاضرات القادمة.